A LIVE COMPUTER SIMULATION OF SWARM DEVELOPMENT IN *SALPA FUSIFORMIS* (TUNICATA : THALIACEA) IN THE FRONTAL ZONE OF THE LIGURIAN SEA (MEDITERRANEAN)

Philippe Laval

CNRS URA 2077 / Université P. et M. Curie, Station zoologique F-06234 Villefranche-sur-Mer, France

Abstract

Self-reproducing artificial creatures are used to simulate the spreading of a salp swarm in space. Each individual is modeled as an autonomous software object, containing its own thread of control. These individuals "live" concurrently and asynchronously: they swim, eat food, metabolize, and reproduce like real salps. The 21 life-cycle and physiological parameters of the creatures were first calibrated with food regularly scattered in all directions (in what I call a virtual mesocosm). The space is then modified to include a process simulating the convergence current carrying organic matter, with food periodically added in the upper layers to simulate phytoplankton growth.

Key-words : models, zooplankton, blooms, particle flux, Ligurian Sea

Introduction

Salps are fragile macroplanktonic organisms, difficult to keep in the laboratory during several generations [1]. Their explosive asexual multiplication allows them to quickly form large swarms which have considerable impact on the pelagic ecosystem. However, their presence or absence is very difficult to predict, and costly investigations (for example with submersibles), which should be planned in advance, may well find only a few specimens [2]. Salp asexual multiplication may be described by analytical mathematical models [3]. However, how a salp swarm is triggered, and how it extends in space, are phenomenons too difficult to write down in mathematical terms.

A software simulation of salp swarms may help to answer the above questions, provided:

1) The software is complicated enough to take into account the existing biological and physiological knowledge about salps;

2) Real-time and concurrency techniques are used to simulate the complex interactions of individuals living simultaneously, but not synchronously;

3) The software has a strong, biologically meaningful, architecture. An ad-hoc design, like the ones often found in object-oriented programs, constitutes a too weak foundation.

Principles of software engineering should be followed from the beginning. These principles tell us: design the architecture first, and then flesh it with the peculiarities of the domain. This is rarely adhered, because biologists are not exposed to software engineering notions during their curriculum. Moreover, concurrent programming is much more difficult than the sequential programs they may have written to compute equations in mathematical models.

Having a biological background in gelatinous macroplankton, and a long experience with the Ada programming language, I have tried since several years to write a simulation software along these lines.

Methods

1) Software design

The software architecture is made along the lines of a software engineering methodology called HOOD (for Hierarchical Object-Oriented Design) [5]. This method considers the domain to be programmed as a hierarchical decomposition in more and more detailed abstract machines. The first level is the whole system itself. It is then split in a few abstract entities, which together provide the same functionalities. This first step is one of the more difficult, because it is necessary to devise logical abstractions which will remain consistent with the subsequent decomposition of their sub parts in the following levels. This usually require a number of iterations to arrive at a stable structure. The decomposition of the child objects stops when an object needs not be further decomposed (terminal object).

2) Artificial salps

Only one artificial oozooid is necessary to initiate a swarm development, because this individual gives birth to a chain of aggregates; each aggregate in the chain gives in turn a new oozooid, and so on. Oozooids and chains are variants of a software module containing several fields keeping local information (birth date, actual position and direction, amount of reserves, etc.). Another field contains a pointer to an Ada task [6]. This task executes all the events making up the life cycle of the individual. It is an asynchronous process running periodically, attached to the particular individual containing the pointer. The cycles of the task correspond to the individual biological clock, running on a "daily" basis. A "day" lasts in fact 1.5 second of computer time, but all actions have proportionally scaled durations relatively to this day length. This permits to observe several generations in less than 300 s.

To visualize the salp population, each individual is displayed with a little colored dot on the computer screen. The screen corresponds to a vertical bi-dimensional grid of 222 x 318 positions, which is the space where the individuals move (for short I call zooids both oozooids and chains when a distinction is not needed). Chains are represented by only one special "individual", because its composing aggregates are clones with the same behavior, about the same position, the same age, etc.

The actions executed by the task are programmed to correspond closely to the biology and physiology of *S. fusiformis*. These actions are parameterized with 21 coefficients or initial values, corresponding to life-cycle characteristics (number of chains emitted by an oozooid, minimum number of aggregates composing a chain, delay for the first chain emission, between-chains interval, oozooids and chains maximum longevity, etc.), or to metabolic parameters (initial and adult weight of oozooids and chains, growth coefficients, metabolic coefficients, etc.). More details are available in [4]. When possible, values were taken from the literature or from laboratory data. If no previous knowledge is available for a parameter, best guesses are made and the results controlled by "dissecting" some individuals:their weight, their amount of reserves may be accessed. Direct observation of the screen provides feedback on the duration of life stages, of the number of oozooids produced by a chain and other data. At the end of the simulation detailed data are recovered by recursively traversing the tree of pointers starting from the first created oozooid.

In fact, the whole system is a model, but more detailed and with a finer spatial resolution than a mathematical model.

3) Environment

In a first stage, the software was calibrated by providing regularly spaced food patches in the grid. I call this setting a "virtual mesocosm" [4]. This step was necessary to verify that the artificial organisms have acceptable physiology and life-cycle.

Once the artificial salps behave convincingly, their environment may be complicated to represent a known ecological situation. The problem of a suitable spatial scale should first be solved. If one dot on the computer screen corresponds to one individual, the resulting scale is too detailed: the software simulates only a small swarm in a reduced volume of water. To extrapolate to a larger water mass, some kind of compromise is unavoidable. It is possible to consider, without too much loss of information, that:

- The actions effected by an individual at a grid position during 1 "day" are average daily actions: feeding is the filtering of the volume at this location during 1 day, metabolism is a mean daily metabolism, and so on.

- In particular, a movement from one position to the next represents the mean distance traveled in a day. For S. fusiformis, this corresponds to about 1,000 m (in projection) in the horizontal direction. In the vertical direction, it is convenient to think that the grid displays only the upper 100 meters. To introduce volumes, the grid has a small depth (not seen in projection) of 0.5 m. Thus, the grid represents a thin vertical slice of 222 x 318 "cells" of 0.5 x 1,000 x 0.5 m = 250 m^3 each. The density of food is computed relatively to this volume.

To structure the physical environment according to the situation prevalent in a frontal zone like the one of the Ligurian Sea, in a preliminary setting I have added another concurrent process. This periodic